TTCANopen

CAN(FD)转以太网

使用说明书(v4.0)

（学习型）



TTCANopen 工作室

目　录

. 产品升级说明 …… 3
1. 产品介绍 …… 4
2. 接口和主要指标 …… 6
3. 设备接口说明 …… 8
4. 转换器种类切换 …… 12
5. 转换器连接与电脑网络配置 …… 13
6. 普通调试模块 …… 16
7. 普通工作模块 …… 23
8. 使用指令配置模块参数 …… 27
9. TTCANopen 调试模块 …… 33
10. TTCANopen 工作模块 …… 41
11. TTCANopen 心跳器模块 …… 45
12. 模块 TTCANopen 特性 …… 50
附录 1. TTCANopen 协议 CAN 标识的分配方法 …… 51
附录 2. TTCANopen 协议设备内部多条指令的发送规则… 57
免责声明…… 59

产品升级说明

1）自 2021 年 8 月 16 日，本产品升级到 v3.3

本次升级新增了采样点的选择项。分别是 75%、80%和 85%左右。用户可以根据情况进行更合理的配置。

2）自 2022 年 1 月 16 日，本产品升级到 v3.5

本次升级，应用户要求提供了 UID 显示，具体当模块 CAN 速率选择为 10K 时，帮助界面会有 96 位的 UID 显示。

进一步完善了对 LINUX 的支持。

4）自 2024 年 8 月 1 日，本产品升级到 v4.0

本次升级，主要新增了四项功能。分别是新增 UID 读取指令；模块时间配置指令；CANFD 数据域不加速发送；6 个 ID 选择过滤功能。具体描述详见附录 3

1. 产品介绍

本品支持CAN和CANFD应用，可做为CAN(FD)转以太网、CAN(FD)转USB、CAN(FD)转TTL模块使用，模块状态切换自如，可最大限度的适用用户的应用场景。

关于CAN(FD)转USB的内容请参考《CAN(FD)转USB使用说明书v3.3》。

关于CAN(FD)转TTL串口的内容请参考《CAN(FD)转TTL使用说明书v3.3》。

本说明书后续主要介绍CAN(FD)转以太网的相关内容。

CAN(FD)转以太网模块主要用于将CAN或CANFD网络通过以太网端口转换使其能够实现“远现场”接入上位监控和管理系统，如：工业现场数据监控等。

其广泛应用于汽车电子，工业自动化监控，医疗仪器，机器人，纺织机械，安防系统，水文气象等领域的数据传输和监控。

TTCANopen CAN(FD)转以太网产品分为三种类型。

第一类：学习型，主要面向青年学生和初学者，转换器主控芯片为GD32E103或STM32G431,以太网单元为w5500硬核

IP，CAN 端口为非隔离方式，集成多种功能，以裸板形式提供，可极大降低学习成本。

第二类：安全型，CAN 端口采用集成磁隔离芯片进行隔离，可有效的保护 PC 端设备，转换器采用工业标准生产和封装，功能同学习型。

第三类：工程型，主要面向工程应用，以单一定制功能模块提供，其它同安全型模块。

本品遵循 CAN 总线协议 2.0A、2.0B 和 ISO11891-1:2015 CANFD 规范，与上位机通讯形式为 TCP 或 UDP，包括服务端模式和客户端模式。（学习型默认为 TCP_server， IP 地址：192.168.1.123 端口号:5000）。支持 CAN 和 CANFD 转以太网。

本品接口通讯能力已经达到相关理论设计值，能够在 6M CANFD 总线 100%满载的条件下，网络端口不会丢帧（见测试报告）。

本着“开放，共享”的应用理念，本品准许用户非商业性 DIY，提供资料中包含设备的“电路逻辑图”“PCB 印制板图”和“ISP 装载固件”，用户可以根据这些文件非常方便的复制出一模一样的作品，从而解除了用户应用的后顾之忧。

2. 接口和主要指标

学习型: USB 端口链接器: A 型链接器

USB 端口通讯: USB2.0 全速， 虚拟串口。

采用 USB 端口+5V 供电

以太网端口：w5500 芯片，RJ45 网络接口

以太网通讯：默认 TCP_server

本机 IP 地址：192.168.1.123 端口号:5000

TTL 端口：三线 TTL 串口（3.3V 供电）

TTL 串口配置： 波特率：256000

奇偶校验：无

数据位：8

停止位：1

CAN 端口链接器: 单排 3P 插针（2.54mm）

CAN 收发器: QBD1044

终端电阻: 平衡式终端电阻

CAN 通讯速率 kbps: 10、20、50、100、125、250、500、800、1000

CANFD 数据速率：为 CAN 速率的整倍数，

最高 6Mbps（或 5Mbps）

核心 ARM 芯片：GD32E103 或 STM32G431

外观尺寸：63×28×30 mm

模块集成：集成三类转换器，支持 CAN 和 CANFD

1. CAN(FD)转以太网
2. CAN(FD)转 USB
3. CAN(FD)转 TTL

在三类转换器中，都集成有五种子模块

1).普通调试模块、

2).普通工作模块、

3).TTCANopen 调试模块、

4).TTCANopen 工作模块、

5).TTCANopen 心跳器模块

工程型:　（具体参考工程型使用说明书）

3. 设备接口说明

1）ISP 固件加载更新接口

本品为用户提供了 ISP 固件加载更新接口，见图 1 所示，用户可以从 www.ttcanopen.com 网站获取产品的最新升级版本，使用 ISP 下载器将最新的固件安装到设备上（当然用户也可以将自己研发的固件安装到设备上）。

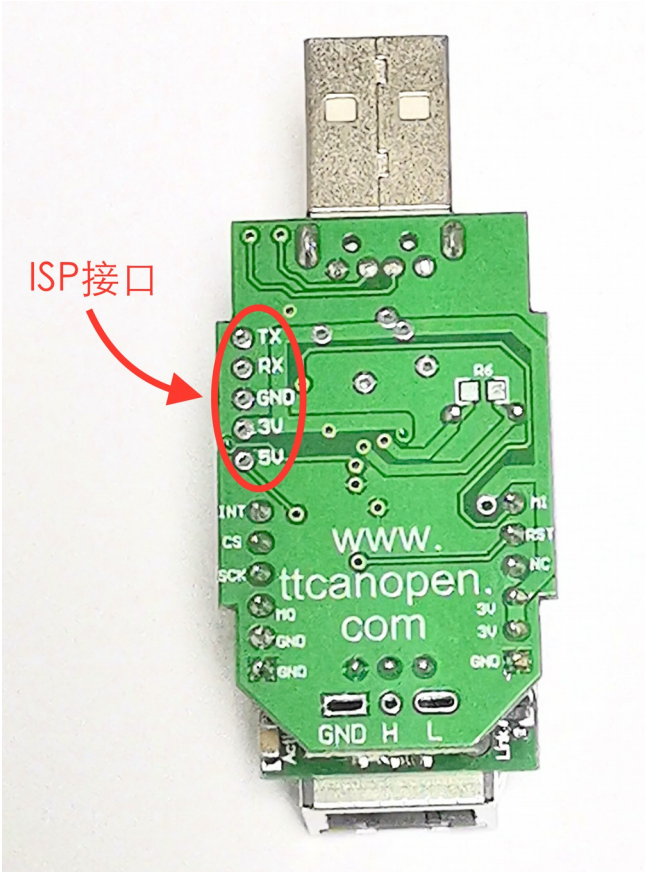


图 1　设备 ISP 固件更新接口

ISP 接点从上至下分别为：TX,RX,GND,3.3V,5V。

下载资料中包含 ISP 运行程序，解压后可以直接运行，请不要擅自更改 ISP 串口配置参数。

用户可以使用市面上多数 ISP 下载器安装设备固件，使用时请注意插针的排序，连接不当可能会损毁设备。

用户也可以到 TTCANopen 淘宝网店购买设备专用的 ISP 下载器，其使用非常方便，只需将专用 ISP 下载器的 5 根插针直接从背面插入到设备的对应 ISP 接口上即可，见图 2 所示。

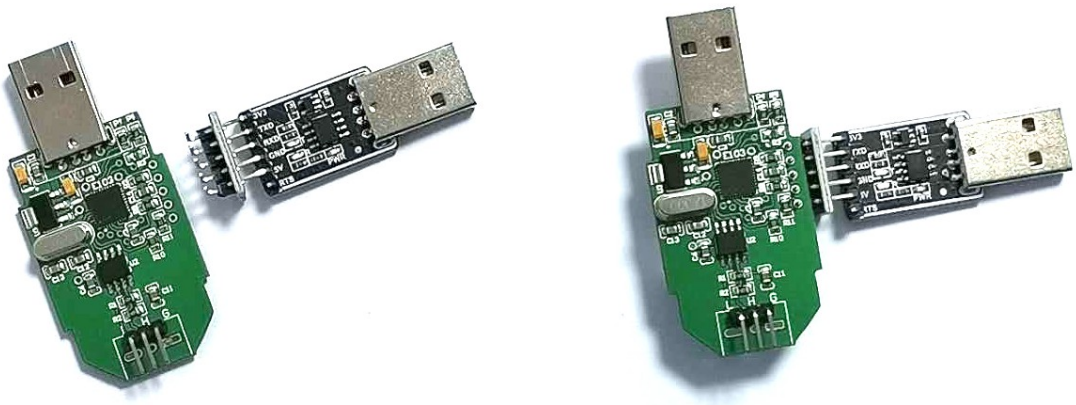


图 2 专用 ISP 下载器的连接

当产品工作在 CAN(FD)转 TTL 串口时（即：不插接 w5500 模块和不连接 USB 接口），设备复用了 ISP 接口的 TX,RX,GND 做为 TTL 串行通讯接口；复用了 ISP 接口的 3.3v,GND 做为设备供电电源接口。

2）以太网 w5500 模块安装接口 SPI

本品以太网接口是使用 w5500 集成模块完成的（兼容 WIZ820io）。其是通过两个单排 6p 孔座连接到主工作板 SPI 接口上的，见图 3 所示。

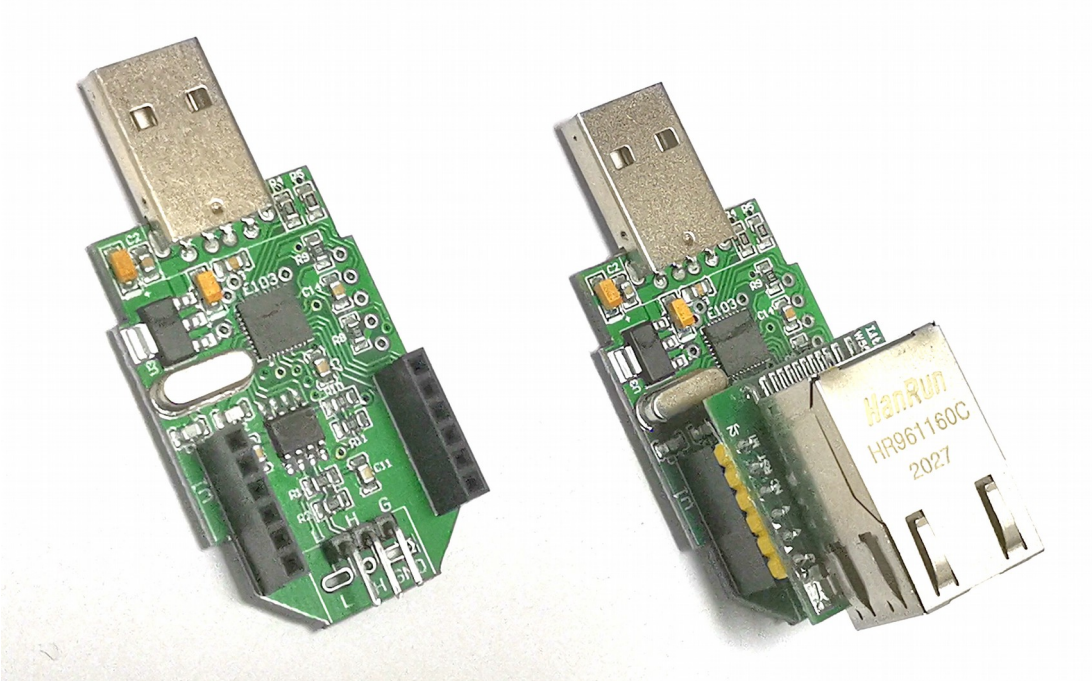


图 3 w5500 模块的安装接口

在插接时，需要特别当心**插接方向**，且**插针不能错位**，否则，上电后容易损毁设备。

另外，此接口还可根据应用需求，扩展为 CAN(FD)转 SPI 使用， 具体请关注 www.ttcanopen.com 网站相关信息。

3）CAN(FD)端口

模块CAN(FD)接口有三个接线端子，见图4所示，分别是CAN_L（左）、CAN_H（中）总线信号和GND（右）地线，通常总线信号线为双绞线，地线可选（多为屏蔽地）。

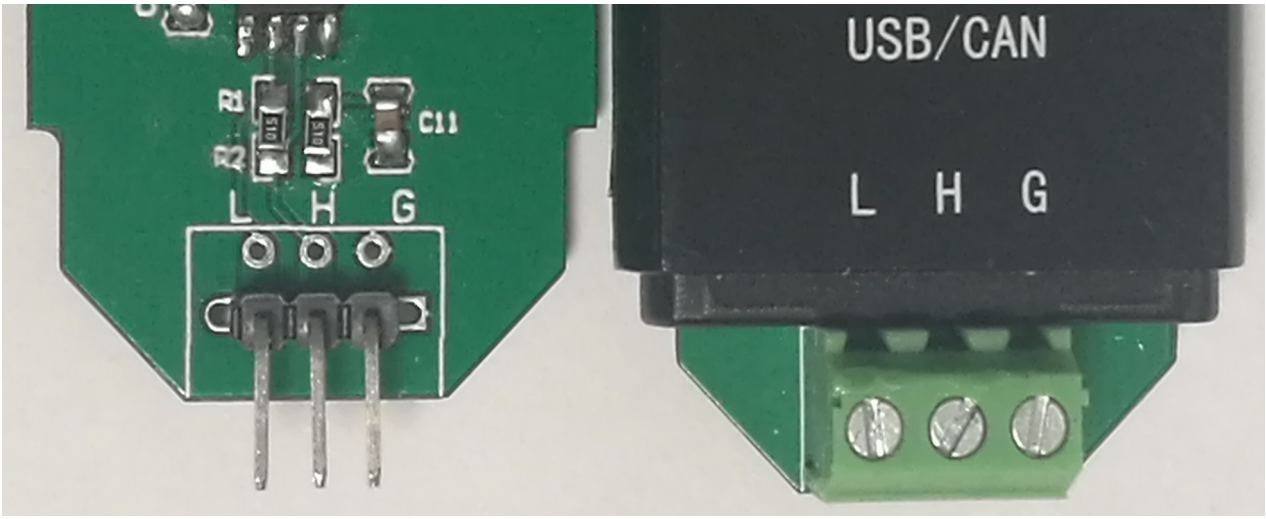


图4 CAN端口接线端子

4. 转换器种类切换

1）当USB端口仅做为供电接口使用时，设备上电默认为CAN(FD)转以太网。

2）当USB端口插入PC机，不打开串口，设备默认仍为CAN(FD)转以太网。

3）当USB端口插入PC机，并打开串口，设备会自动切换为CAN(FD)转USB。

4）当USB串口通讯波特率设置为192168时，设备会切换回CAN(FD)转以太网。

5）当不安装w5500以太网模块时，设备仍然可以做为CAN(FD)转USB工作。

6）当不安装w5500以太网模块，同时也不接入USB端口，设备复用ISP端口的3.3v和GND供电，复用ISP端口的TX,RX,GND做为TTL串行通讯接口，设备自动配置为CAN(FD)转TTL串口。

设备上电后，CAN(FD)端口默认为“静听状态”，CAN速率为1000K，FD速率为2000K。用户需要修改这个配置，以使其与当前应用CAN环境相匹配。

5. 转换器连接与电脑网络配置

1）转换器可以使用网线直接与 PC 机的网口连接；见图 5

2）转换器也可以与 HUB 或者 WIFI 路由器连接，PC 机可采用有线或无线的方式接入 HUB 或 WIFI；见图 6，图 7，所示。

注意：以太网接口同时只支持一个客户端接入。

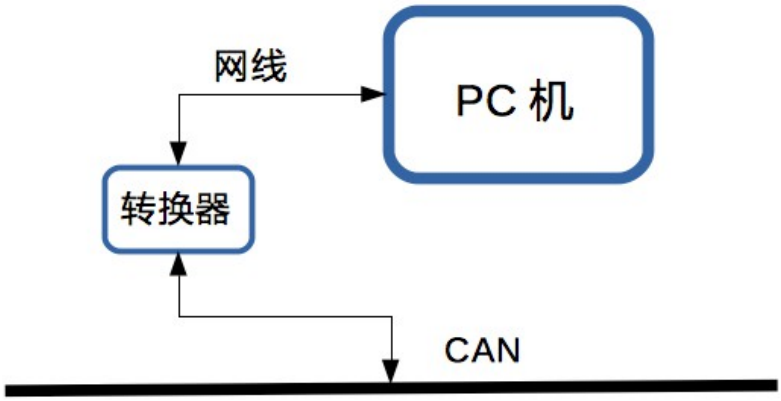


图 5 转换器直接与 PC 连接

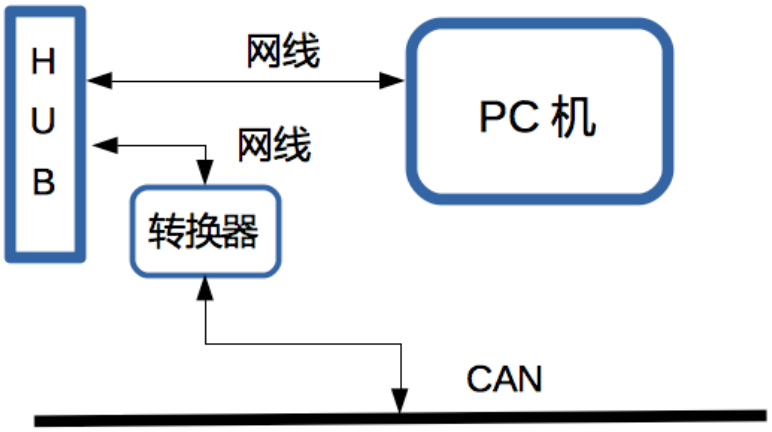


图 6 转换器与 HUB 连接

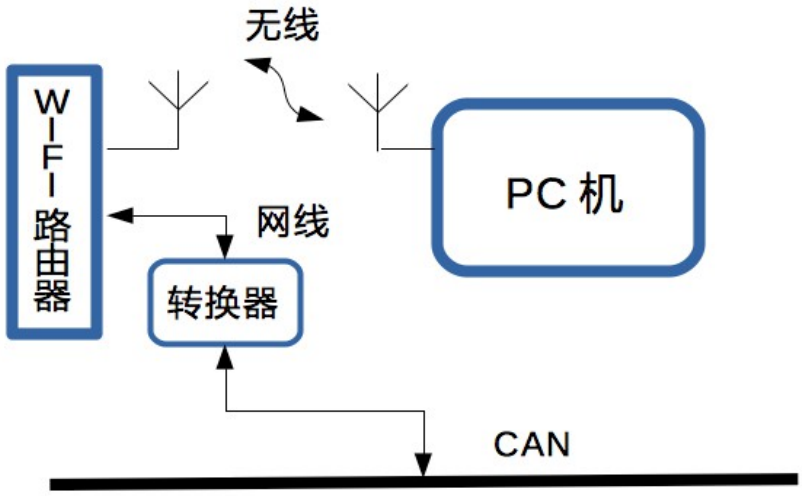


图 7 转换器与 WIFI 路由器连接

客户端 PC 机 IP 地址配置应当属于 192.168.1.XXX 子网，见图 8 所示。

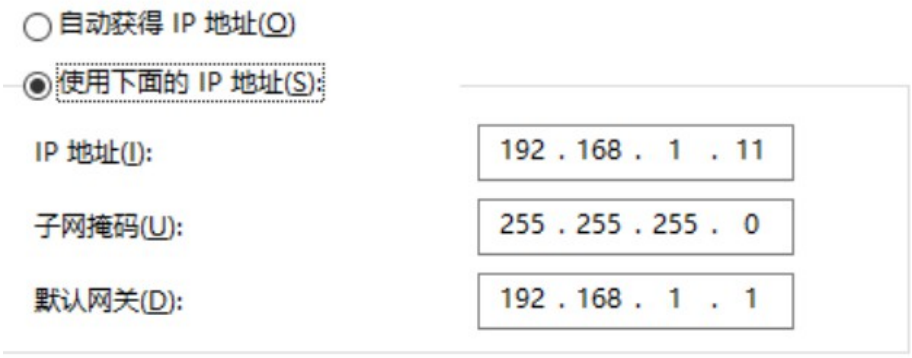


图 8 客户端 IP 地址设置

转换器连接和 PC 机配置好后，并对应设置好网络调试助手上的相关参数，点击调试助手的“连接”按钮，连接成功，在接收窗口会出现转换器的帮助信息和当前 CAN(FD)接口的状

态信息，见图9所示。

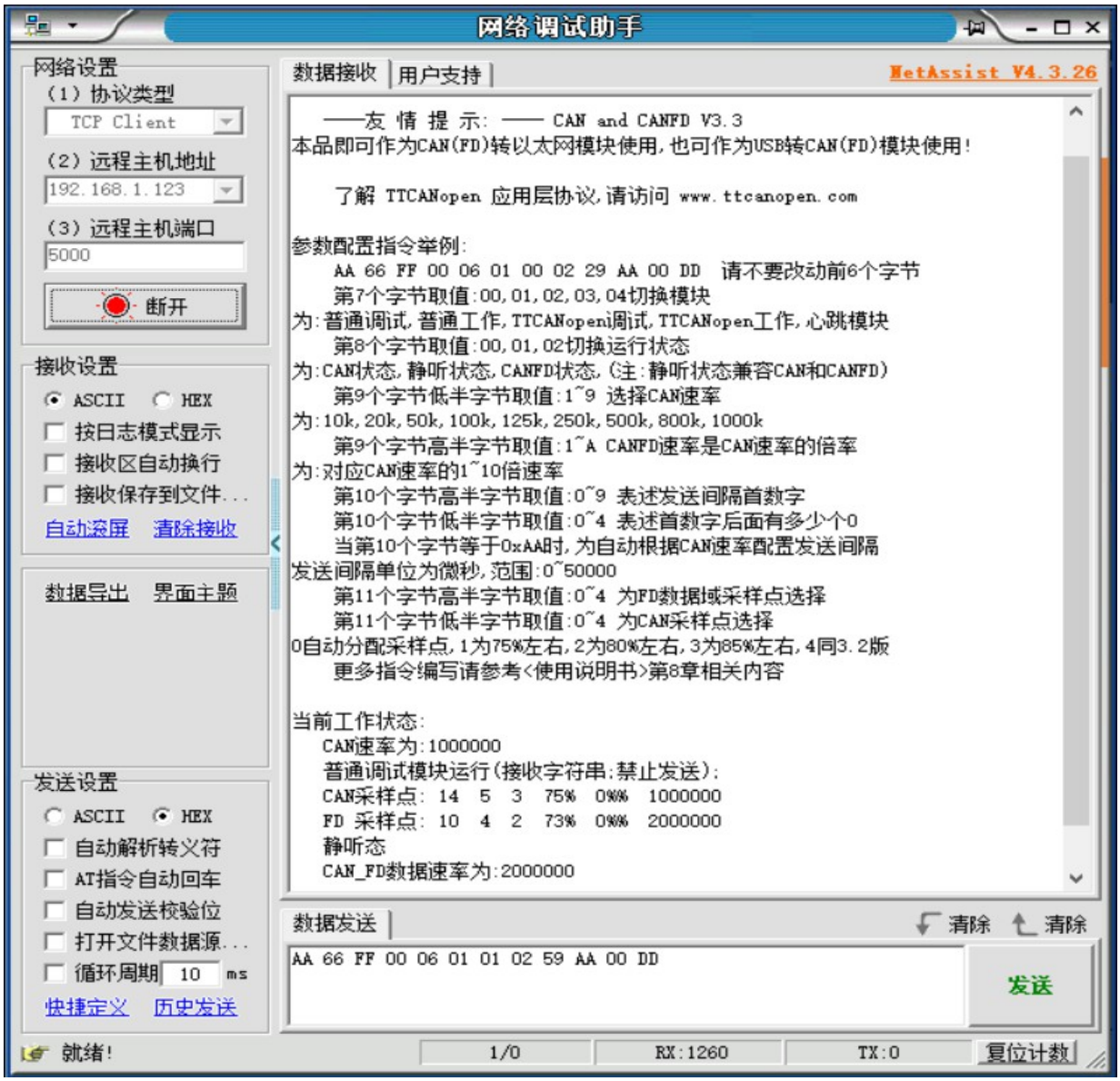


图9 转换器的帮助信息和当前状态

用户需要使用“配置指令”（详见第八节内容），重新配置CAN(FD)端口参数，以使其与当前应用环境相匹配。

6. 普通调试模块

用户可以通过配置指令，将设备配置为普通调试模块。配置方法见本《说明书》第 8 节。

普通调试模块，是用户在调试已有 CAN 或 CANFD 网络时，临时加入的调试设备，待调试完成后脱离原网络。其不是原 CAN 网络的组成部分。

用户可以将模块配置为 CAN 状态或 CANFD 状态，以使其与被调试的 CAN 网络状态相匹配，当用户只需对原网络进行监听时，也可将模块配置为静听态，这样对原网络的影响最小，并能同时静听原网络中的 CAN 指令和 CANFD 指令。

为了方便用户观察原网络中指令的运行情况，普通调试模块输出为字符串格式，每帧指令独占一行，用户可使用网络调试助手进行调试观察。

普通调试模块支持 4 种指令帧的显示，其格式如下：

1）显示标准遥控帧指令格式：

帧计数	接收时标	帧标识	ID 字	DLC	帧标识
00000000 至 FFFFFFFF	时:分:秒:毫秒.微妙	<	0000 至 07EF	0 至 8	>

2）显示标准数据帧指令格式：

帧计数	接收时标	帧标识	ID 字	DLC	帧标识	数据场 DLC 个字节
00000000 至 FFFFFFFF	时:分:秒:毫秒.微妙	[	0000 至 07EF	0 至 8	]	xx ～ xx

3）显示扩展遥控帧指令格式：

帧计数	接收时标	帧标识	ID 字	DLC	帧标识
00000000 至 FFFFFFFF	时:分:秒:毫秒.微妙	<	00000000 至 1FBFFFFF	0 至 8	>

4）显示扩展数据帧指令格式：

帧计数	接收时标	帧标识	ID 字	DLC	帧标识	数据场 DLC 个字节
00000000 至 FFFFFFFF	时:分:秒:毫秒.微妙	[	00000000 至 1FBFFFFF	0 至 8	]	xx ～ xx

注：上述格式是按普通 CAN 状态表述的，CANFD 状态不支持遥控帧，CANFD 指令中 DLC 字段和其所对应的数据场长度，遵守 CANFD 协议相关规定。DLC 字段用 10 进制显示。

普通调试模块接收指令显示举例：

1）标准遥控帧：

00000001 00:03:07:671.200 < 0123 8 >

2）标准数据帧：

00000002 00:03:07:671.346 [0321 4] 11 22 33 44

3）扩展遥控帧：

00000003 00:03:07:671.443 < 12345678 6 >

4）扩展数据帧：

00000004 00:03:07:671.605 [18765432 3] 11 22 33

上例中:（举例:扩展数据帧）

第一个字符串“00000004”：

为接收帧计数，计数范围 0～0xFFFFFFFF;

第二个字符串“00:03:07:671.605”：

为接收时标，时：分：秒：毫秒.微妙，分辨率为 1 微秒;

第三个[扩起来]的字符串“[18765432 3]”：

为帧 ID 和 DLC，(ID 为 32 位十六进制数,低 29 位有效。DLC 为十进制数，CAN 状态为 0 至 8，CANFD 状态为 0 至 64，并遵守 CANFD 协议相关规定)；<尖括号>则表述遥控帧;

（注：对于标准帧，ID 为 16 位十六进制数，低 11 位有效）

第四个字符串“11 22 33 ”：

为数据帧中 DLC 个字节数据内容。

图 10 为在网络调试助手观察到的四种帧举例，分别为：标准遥控帧、标准数据帧、扩展遥控帧和扩展数据帧。



图 10 普通调试模块接收显示举例

普通调试模块支持4种指令帧的发送，并对发送帧进行“回显”，发送指令使用十六进制方式，其格式如下：

1）发送标准遥控帧指令格式：

帧头	帧标识	ID字	DLC	帧尾
AA	77	0000 至 07EF	00 至 08	DD

2）发送标准数据帧指令格式：

帧头	帧标识	ID字	DLC	数据场DLC个字节	帧尾
AA	55	0000 至 07EF	00 至 08	xx ～ xx	DD

3）发送扩展遥控帧指令格式：

帧头	帧标识	ID字	DLC	帧尾
AA	FF	00000000 至 1FBFFFFF	00 至 08	DD

4）发送扩展数据帧指令格式：

帧头	帧标识	ID字	DLC	数据场DLC个字节	帧尾
AA	CC	00000000 至 1FBFFFFF	00 至 08	xx ～ xx	DD

注：CANFD 不支持遥控帧，CANFD 指令中 DLC 字段和其所对应的数据场长度，遵守 CANFD 协议相关规定。

普通调试模块发送指令举例（十六进制方式）：

1）发送标准遥控帧：

AA 77 0123 08 DD

2）发送标准数据帧：

AA 55 0321 08 11 22 33 44 55 66 77 88 DD

3）发送扩展遥控帧：

AA FF 12345678 06 DD

4）发送扩展数据帧：

AA CC 18765432 08 11 22 33 44 55 66 77 88 DD

图 11 为网络调试助手中普通调试模块发送指令和回显示例，分别为：标准遥控帧、标准数据帧、扩展遥控帧和扩展数据帧。

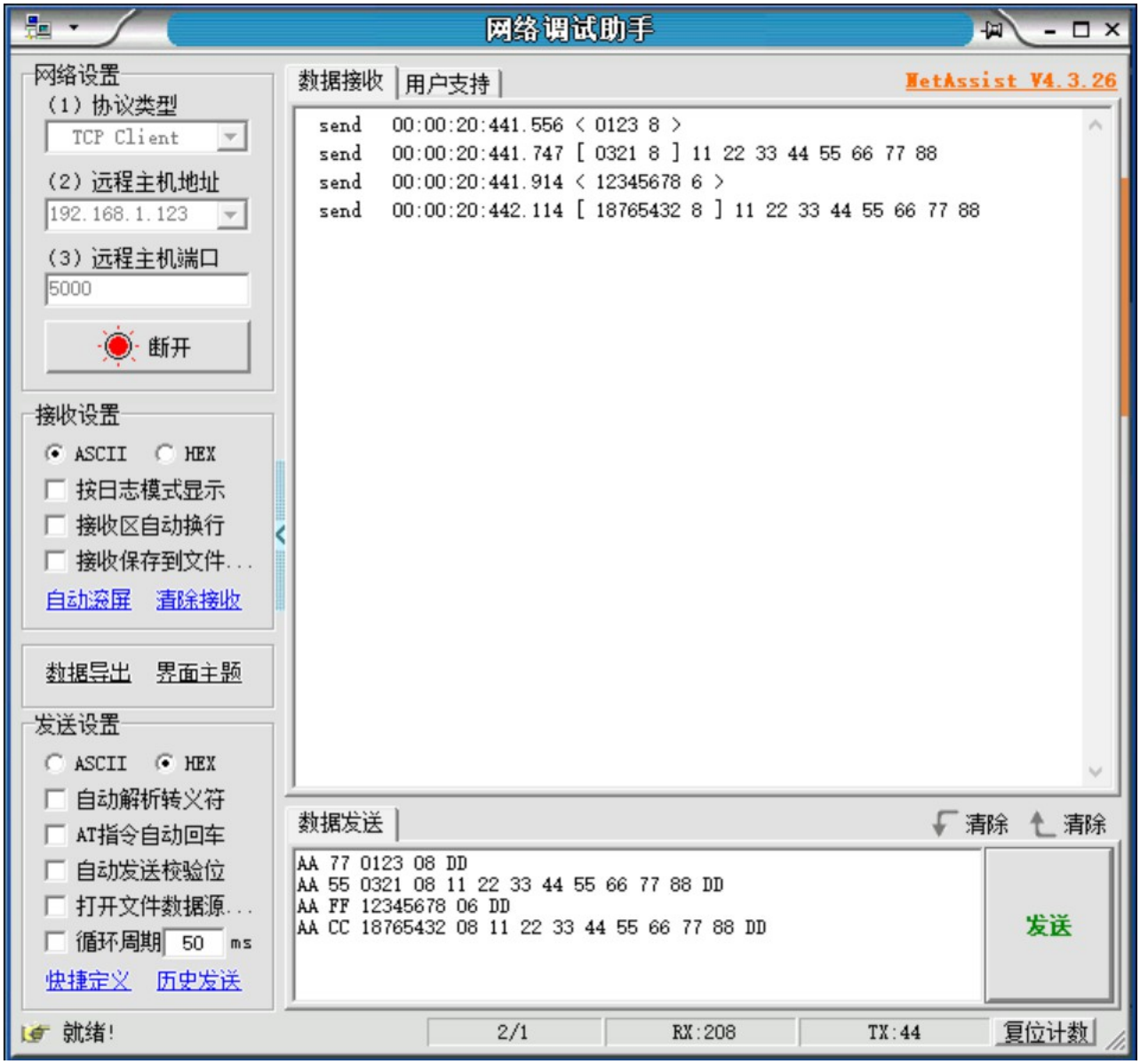


图 11 普通调试模块发送指令和回显示例

7. 普通工作模块

用户可以通过配置指令，将设备配置为普通工作模块。配置方法见本《说明书》第 8 节。

普通工作模块，是已有 CAN 或 CANFD 网络原生的组成部分，其主要承担 CAN 网络与系统上位机的通讯连接和协议转换，是中心监控系统的交通枢纽。

用户可以将模块配置为 CAN 状态或 CANFD 状态，以使其与用户当前 CAN 网络状态保持一致。

为了提高指令效率，工作模块输入输出皆使用 16 进制格式，帧与帧之间使用帧头帧尾标识。

普通工作模块支持 4 种指令帧的输入输出，其中，发令格式与普通调试模块相同，但普通工作模块不“回显”发送指令。

普通工作模块接收指令格式如下：

1）接收标准遥控帧指令格式：

帧头	接收时标	帧标识	ID 字	DLC	帧尾
BB	tt tt tt tt	77	0000 至 07EF	00 至 08	EE

2）接收标准数据帧指令格式：

帧头	接收时标	帧标识	ID 字	DLC	数据场 DLC 个字节	帧尾
BB	tt tt tt tt	55	0000 至 07EF	00 至 08	xx ～ xx	EE

3）接收扩展遥控帧指令格式：

帧头	接收时标	帧标识	ID 字	DLC	帧尾
BB	tt tt tt tt	FF	00000000 至 1FBFFFFF	00 至 08	EE

4）接收扩展数据帧指令格式：

帧头	接收时标	帧标识	ID 字	DLC	数据场 DLC 个字节	帧尾
BB	tt tt tt tt	CC	00000000 至 1FBFFFFF	00 至 08	xx ～ xx	EE

注 1：tt tt tt tt 为 32 位接收时间计数（计数单位为 0.1ms），其低字节在前，高字节在后；而 ID 字是高字节在前，低字节在后，解帧时需特别留意。

注 2：CANFD 不支持遥控帧，CANFD 指令中 DLC 字段和其所对应的数据场长度，遵守 CANFD 协议相关规定。

普通工作模块接收指令举例：

1）标准遥控帧：

BB B5 3C 02 00 77 01 23 08 EE

2）标准数据帧：

BB B6 3C 02 00 55 03 21 08 11 22 33 44 55 66 77 88 EE

3）扩展遥控帧：

BB B7 3C 02 00 FF 12 34 56 78 06 EE

4）扩展数据帧：

BB B9 3C 02 00 CC 18 76 54 32 08 11 22 33 44 55 66 77 88 EE

图 12 为普通工作模块，指令接收示例，分别为：标准遥控帧、标准数据帧、扩展遥控帧和扩展数据帧，在这里每帧指令并没有独立成行，而是用帧头 BB 和帧尾 EE 进行分割的。

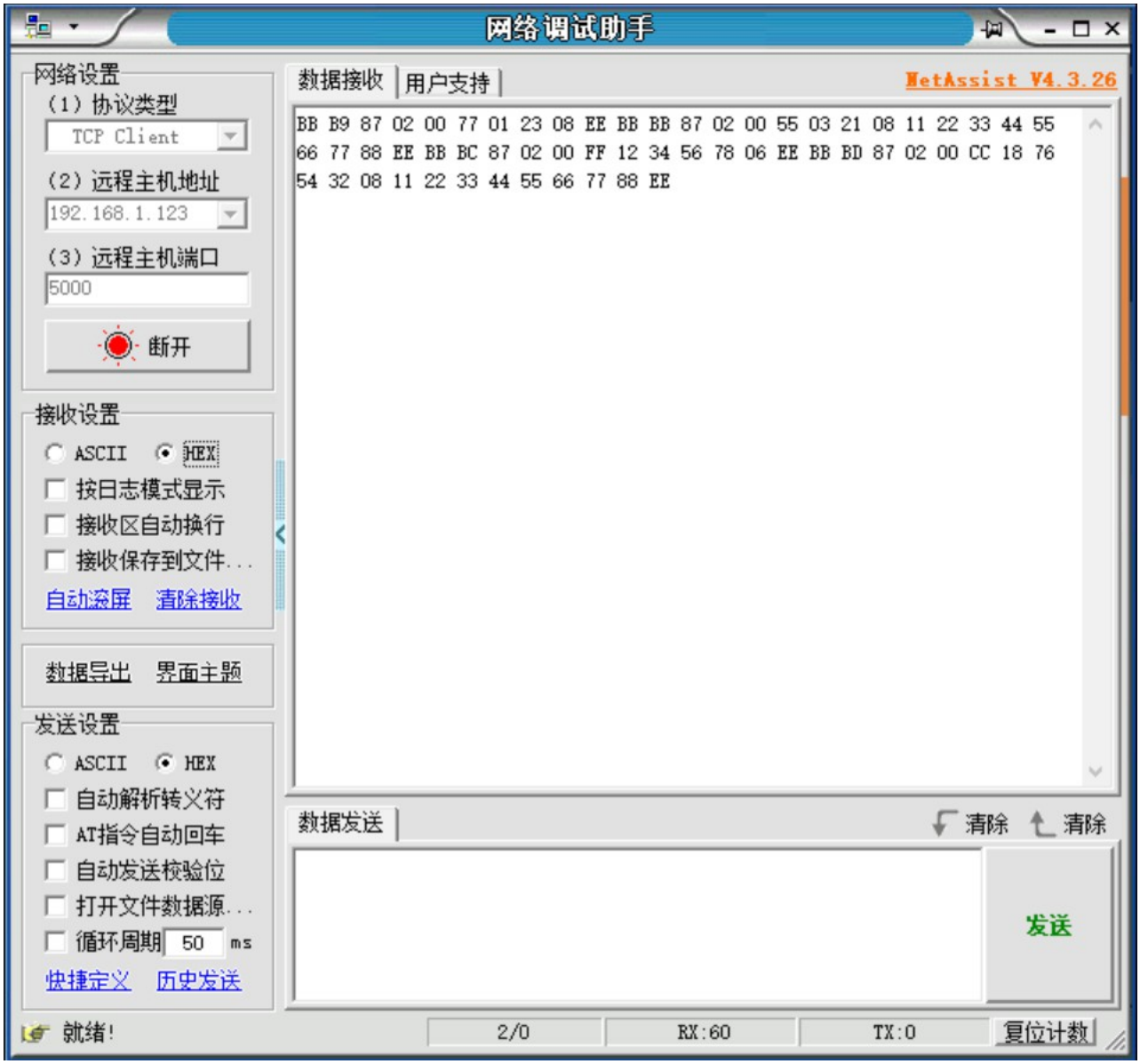


图 12 普通工作模块接收指令示例

8. 使用“指令”配置模块参数

设备上电后，其CAN(FD)端口的配置参数往往与当前应用环境不一致，这就需要使用专用的“配置指令”重新设置设备相应的配置参数，以使其与当前应用环境相匹配。

实际上，在设备内部分配了一个固有的寄存器空间来存放其配置参数，用户可以通过修改该寄存器中的内容，完成配置参数和设置工作状态。表1为该寄存器中相关参数的分配状况。

表1 固有的配置参数寄存器空间的分配

序号	寄存器地址	字节数	参数描述
1	FF00	1	当该值置1时，启动配置，然后自动归0.
2	FF01	1	选择工作模块： 0:为普通调试模块 1:为普通工作模块 2:为TTCANopen调试模块 3:为TTCANopen工作模块 4:为TTCANopen心跳模块
3	FF02	1	配置工作状态： 0:为CAN状态 1:为静听态 2:为CANFD态
4	FF03	1	配置CAN和CANFD速率： 低半字节配置CAN速率

			1～9 分别对应 CAN 速率的 10、20、50、100、125、250 、500、800、1000 kbps 高半字节配置 CANFD 的倍率 1～A 分别对应 CANFD 为 CAN 速率的 1～10 倍
5	FF04	1	设置两帧发送间隔（微妙） 高半字节为发送间隔首数据， 低半字节为 0 的个数。 注意：当该字节被设置为 0xAA 时 将由模块根据通讯速率自动 生成发送间隔
6	FF05	1	选择配置采样点 低半字节配置 CAN 采样点 高半字节配置 FD 采样点 0:为自动配置采样点 1:为 75%左右 2:为 80%左右 3:为 85%左右 4:同 v3.2 版
7	FF06～FF0F	2	保留

为了能够设置和修改该寄存器内容，设计了专用指令格式如下：

设置配置参数寄存器指令格式：

帧头	帧标识	寄存器地址	字节数	数据内容	帧尾
AA	66	FF00 至 FF0F	01 至 10	xx ～ xx	DD

用户配置完参数后，需将 0xFF00 寄存器置 1，新配置方才生效，生效后该寄存器内容自动归零。

配置参数指令举例：

例 1；修改 CAN 发送速率为 1000000bps

AA 66 FF03 01 19 DD

AA 66 FF00 01 01 DD

例 2:修改 CAN 工作状态为 CANFD

AA 66 FF02 01 02 DD

AA 66 FF00 01 01 DD

当然用户也可以使用一条指令将配置参数一次性完成（推荐）。

例 3:一次配置全部参数，包括配置立即生效

AA 66 FF00 06 01 01 02 59 AA 00 DD

该指令配置为“普通工作模块”、“CANFD 态”、“CAN 速率 1000kbps”、“CAN FD 速率 5000kbps”、“自动配置发送间隔”、“自动配置采样点”配置立即生效。

当配置成功后，接收窗口会有信息显现见图 13.

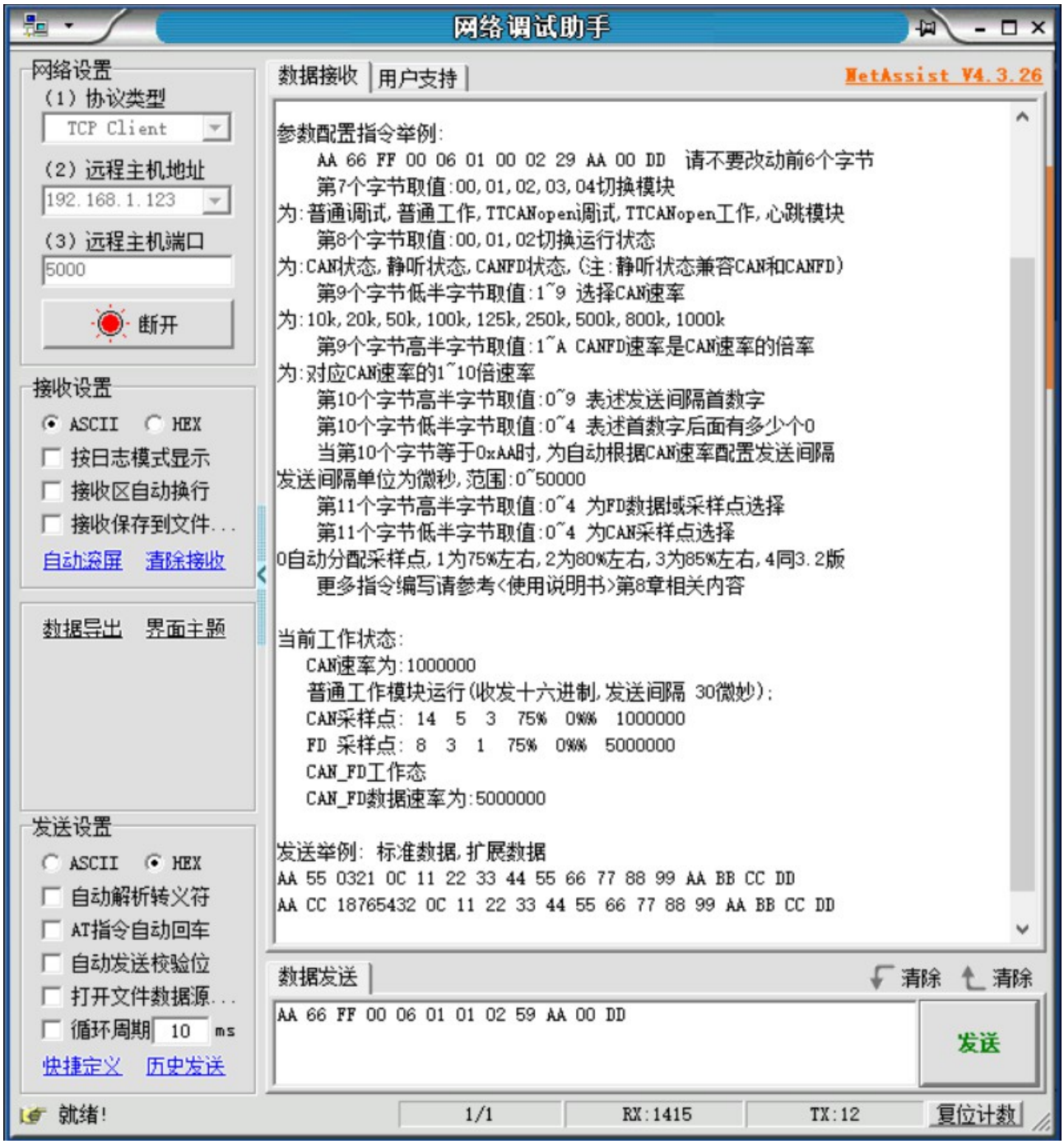


图 13 使用配置指令成功显示反馈信息举例

用户在配置模块参数时，需要注意可配置参数的取值范围以其适用性，如：通讯速率除了受主控 MCU 芯片的限制，同时也受 CAN 接口芯片指标的限制，以及终端匹配电阻的连接形式的影响。用户在使用模块时应当按照所用模块的硬件配置和应用环境，进行合理设置。 表 3 为主控芯片和接口芯片 CAN 通讯速率相关参数：

表 3 模块所用各芯片的 CAN 通讯速率范围

芯片代号	CAN 速率 kbps	FD 速率 kbps
GD32E103	10 至 1000	10 至 6000
STM32H750	10 至 1000	10 至 5000
TJA1057	40 至 1000	40 至 5000
ADM3057	10 至 1000	10 至 12000
QBD1044	10 至 1000	10 至 6000

另外，需要提醒用户的是对于定制“工程模块”，其配置参数和工作状态是用户定制的，当模块上电后，定制参数立即生效。

说明书以下部分与TTCANopen协议有关，有需求的用户可以继续阅读。

本品为业界首个支持TTCANopen应用层协议的CAN(FD)转以太网模块，用户可以访问www.ttcanopen.com网站获得更多的TTCANopen相关知识。

TTCANopen是由国人打造的一款拥有自主知识产权的CAN应用层协议，与以往的应用层协议不同， TTCANopen 实际上是一个开放的协议框架。它是由众多的“基本子协议”和“应用子协议”构成，用户可以根据自己的专业应用特点，去选择“应用子协议”;也可以使用“基本子协议”去生成新的“应用子协议”; TTCANopen协议在应用层界面直接兼容最新的CAN_FD。

TTCANopen 应用层协议以连续寄存器空间为信息载体，使用了两个不同的CAN标识符划分的方式分别传递过程变量和非过程变量，巧妙的解决了寄存器空间分层管理与其指令优先级分配不平衡的矛盾。

TTCANopen还特别规范了设备内部多条指令的发送原则，这是其它应用层协议所不具备的。

TTCANopen引入了系统心跳的概念，用以同步总线上的设备，使系统拥有统一的时钟，并使用时间相位矩阵取代了传统的指令查询矩阵，提高了总线带宽的利用率，大大的减小了浪涌出现的概率，使应用系统实现了由传统的指令驱动到现代的时间驱动的跨越。

TTCANopen应用层协议对CAN标识符的划分及其物理含义的分配，如:优先级段、寄存器基地址段、设备编号段、功能码段都具有明确的应用层协议指令含义，据此设计的应用层协议具有很好的直读性，易于解析，大大降低了学习入门的难度。是一款非常适合推广和普及的应用层协议。

另外，TTCANopen有着类modbus特性，使其能够非常容易的和市面上大多数组态软件完美对接。为用户构建应用系统提供了极大的便利。

9. TTCANopen 调试模块

用户可以通过配置指令，将设备配置为 TTCANopen 调试模块。配置方法见本《说明书》第 8 节。

TTCANopen 调试模块，是用户在调试已有 TTCANopen 网络时，临时加入的调试设备，待调试完成后脱离原网络。其不是原 TTCANopen 网络的组成部分。

用户可以将模块配置为 CAN 状态或 CANFD 状态，以使其与被调试的 TTCANopen 网络状态相匹配，当用户只需对原网络进行监听时，也可将模块配置为静听态，这样对原网络的影响最小，并能同时静听原网络中的 CAN 指令和 CANFD 指令。

为了方便用户观察原网络中指令的运行情况，TTCANopen 调试模块输出为字符串格式，每帧指令独占一行，用户可使用网络调试助手进行调试观察。

与普通模块不同，TTCANopen 模块将 29 位 ID 字拆分为四个具有实际物理含义的子段，分别为：优先级段、寄存器基地址段、设备编号段和功能码段，使其能够贴近实际应用构成一套完整的应用层指令系统和管理系统。

TTCANopen 调试模块支持 4 种 TTCANopen 指令帧的显示，其格式如下：

1）显示 TTCANopen 扩展数据帧指令格式：（在用）

帧计数	接收时标	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	帧标识
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	[	0或1	0000至FBFF	00至7F	00至1F	00至08	]

数据场 DLC 个字节
xx ～ xx

2）显示 TTCANopen 扩展遥控帧指令格式：（未用）

帧计数	接收时标	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	帧标识
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	<	0或1	0000至FBFF	00至7F	00至1F	00至08	>

3）显示 miniTTCANopen 标准数据帧指令格式：（未用）

帧计数	接收时标	帧标识	寄存器地址	段地址	DLC	帧标识	数据场 DLC 字节
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	[	00至FD	00至07	00至08	]	xx ～ xx

4）显示 miniTTCANopen 标准遥控帧指令格式：（未用）

帧计数	接收时标	帧标识	寄存器地址	段地址	DLC	帧标识
00000000至FFFFFFFF	时:分:秒:毫秒.微妙	<	00至FD	00至07	00至08	>

注 1：上述格式是按普通 CAN 状态表述的，CANFD 状态不支持遥控帧，CANFD 指令中 DLC 字段和其所对应的数据场长度，遵守 CANFD 协议相关规定。DLC 字段用 10 进制显示。

注 2：对于标准帧，支持 miniTTCANopen 协议，尚未推出。

注 3：TTCANopen 应用层协议只使用了 29 位扩展数据帧。

TTCANopen 调试模块显示帧举例：

1）TTCANopen 扩展数据帧：（在用）

00000012 23:18:56:566.211 [1 7100 32 19 2] 11 22

2）TTCANopen 扩展遥控帧：（注：未启用）

00000013 23:18:56:566.623 < 1 7100 32 19 8 >

3）miniTTCANopen 标准数据帧：（注：未启用）

00000014 23:18:56:567.135 [35 3 4] 11 22 33 44

4）niniTTCANopen 标准遥控帧：（注：未启用）

00000015 23:18:56:567.376 < 35 3 8 >

上例中:（TTCANopen 扩展数据帧）

第一个字符串“00000012”：

为接收帧计数，计数范围 0～0xFFFFFFFF；

第二个字符串“23:18:56:566.211”：

为接收时标，分辨率为 1 微秒；

第三个[扩起来]的字符串“[1 7100 32 19 2]”：

分别为 1 位优先级、16 位寄存器基地址、7 位设备地址、5 位指令功能码和 DLC；（DLC 为十进制数，CAN 状态其为 0 至 8，CAN_FD 状态为 0 至 64，并遵守 CAN_FD 协议相关规定）<尖括号>则表述遥控帧；

第四个字符串“11 22”：

为数据帧中 DLC 个字节数据内容。

图 14 为在网络调试助手接收窗口观察到的四种帧举例，分别为：miniTTCANopen 标准遥控帧、miniTTCANopen 标准数据帧、TTCANopen 扩展遥控帧和 TTCANopen 扩展数据帧。



图 14 TTCANopen 调试模块接收显示举例

TTCANopen 调试模块支持 4 种指令帧的发送，并对发送帧进行“回显”，发送指令使用十六进制方式，其格式如下：

1）发送 TTCANopen 扩展数据帧指令格式：（在用）

帧头	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	数据场 DLC 个字节	帧尾
AA	CC	00 或 01	0000 至 FBFF	00 至 7F	00 至 1F	00 至 08	xx ～ xx	DD

2）发送 TTCANopen 扩展遥控帧指令格式：（未用）

帧头	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	帧尾
AA	FF	00 或 01	0000 至 FBFF	00 至 7F	00 至 1F	00 至 08	DD

3）发送 miniTTCANopen 标准数据帧指令格式：（未用）

帧头	帧标识	寄存器地址	段地址	DLC	数据场 DLC 个字节	帧尾
AA	55	00 至 FD	00 至 07	00 至 08	xx ～ xx	DD

4）发送 miniTTCANopen 标准遥控帧指令格式：（未用）

帧头	帧标识	寄存器地址	段地址	DLC	帧尾
AA	77	00 至 FD	00 至 07	00 至 08	DD

注 1：寄存器基地址为 16 位十六进制数，高地址在前，低地址在后。支持标准帧的 miniTTCANopen 协议格式尚未推出。

注 2：CANFD 不支持遥控帧，CANFD 指令中 DLC 字段和其所对应的数据场长度，遵守 CANFD 协议相关规定。

TTCANopen 发送指令举例：

1）扩展数据帧：（注：TTCANopen 协议只使用扩展数据帧）

AA CC 01 7100 32 19 08 11 22 33 44 55 66 77 88 DD

2）扩展遥控帧：（注：未启用）

AA FF 01 7100 32 19 08 DD

3）标准数据帧：（注：未启用）

AA 55 35 03 08 11 22 33 44 55 66 77 88 DD

4）标准遥控帧：（注：未启用）

AA 77 35 03 08 DD

图 15 为网络调试助手中 TTCANopen 调试模块发送指令和回显示例，分别为：miniTTCANopen 标准遥控帧、miniTTCANopen 标准数据帧、TTCANopen 扩展遥控帧和 TTCANopen 扩展数据帧。

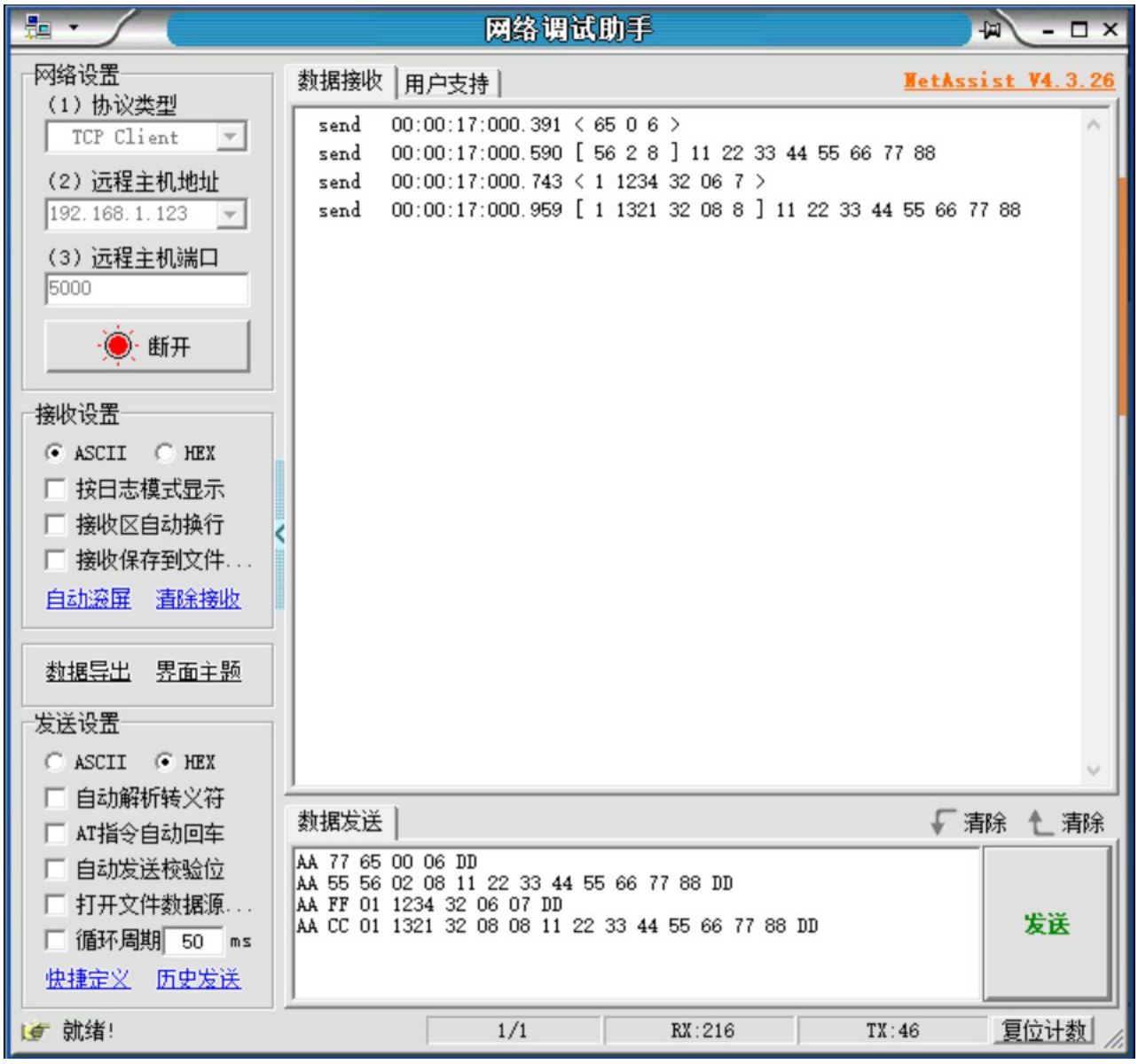


图 15 TTCANopen 调试模块发送指令和回显示例

10. TTCANopen 工作模块

用户可以通过配置指令，将设备配置为 TTCANopen 工作模块。配置方法见本《说明书》第 8 节。

TTCANopen 工作模块，是已有 TTCANopen 网络原生的组成部分，其主要承担 TTCANopen 网络与系统上位机的通讯连接和协议转换，是中心监控系统的交通枢纽。利用 TTCANopen 协议“组态王”驱动，用户可快速建立起自己的中心监控系统。

用户可以将模块配置为 CAN 状态或 CANFD 状态，以使其与用户当前 CAN 网络状态保持一致。

为了提高指令效率，TTCANopen 工作模块输入输出皆使用 16 进制格式，帧与帧之间使用帧头帧尾标识。

TTCANopen 工作模块支持 4 种指令帧的输入输出，其中，发令格式与 TTCANopen 调试模块相同，但 TTCANopen 工作模块不“回显”发送指令。

TTCANopen 工作模块接收格式如下：

1）接收 TTCANopen 扩展数据帧指令格式：（在用）

帧头	接收时标	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	数据场 DLC 个字节	帧尾
BB	tt tt tt tt	CC	00 或 01	0000 至 FBFF	00 至 7F	00 至 1F	00 至 08	xx ～ xx	EE

2）接收 TTCANopen 扩展遥控帧指令格式：（未用）

帧头	接收时标	帧标识	优先级	寄存器基地址	设备地址	指令	DLC	帧尾
BB	tt tt tt tt	FF	00 或 01	0000 至 FBFF	00 至 3F	00 至 1F	00 至 08	EE

3）接收 miniTTCANopen 标准数据帧指令格式：（未用）

帧头	接收时标	帧标识	寄存器地址	段地址	DLC	数据场 DLC 个字节	帧尾
BB	tt tt tt tt	55	00 至 FD	00 至 07	00 至 08	xx ～ xx	EE

4）接收 miniTTCANopen 标准遥控帧指令格式：（未用）

帧头	接收时标	帧标识	寄存器地址	段地址	DLC	帧尾
BB	tt tt tt tt	77	00 至 FD	00 至 07	00 至 08	EE

注 1：tt tt tt tt 为 32 位接收时间计数（0.1ms），其低字节在前，高字节在后；而 ID 字是高字节在前，低字节在后，解帧时需特别留意。

注 2：CANFD 不支持遥控帧，CANFD 指令中 DLC 字段和其所对应的数据场长度，遵守 CANFD 协议相关规定。

在串口调试工具中接收指令举例：

1）TTCANopen 扩展数据帧：（在用）

BB 15 23 00 00 CC 01 71 00 32 19 03 11 22 33 EE

2）TTCANopen 扩展遥控帧：（注：未启用）

BB 16 23 00 00 FF 01 71 00 32 19 08 EE

3）miniTTCANopen 标准数据帧：（注：未启用）

BB 18 23 00 00 55 35 03 06 11 22 33 44 55 66 EE

4）miniTTCANopen 标准遥控帧：（注：未启用）

BB 19 23 00 00 77 35 03 08 EE

图 16 为 TTCANopen 工作模块，指令接收示例，在这里每帧指令并没有独立成行，而是用帧头 BB 和帧尾 EE 进行分割的。

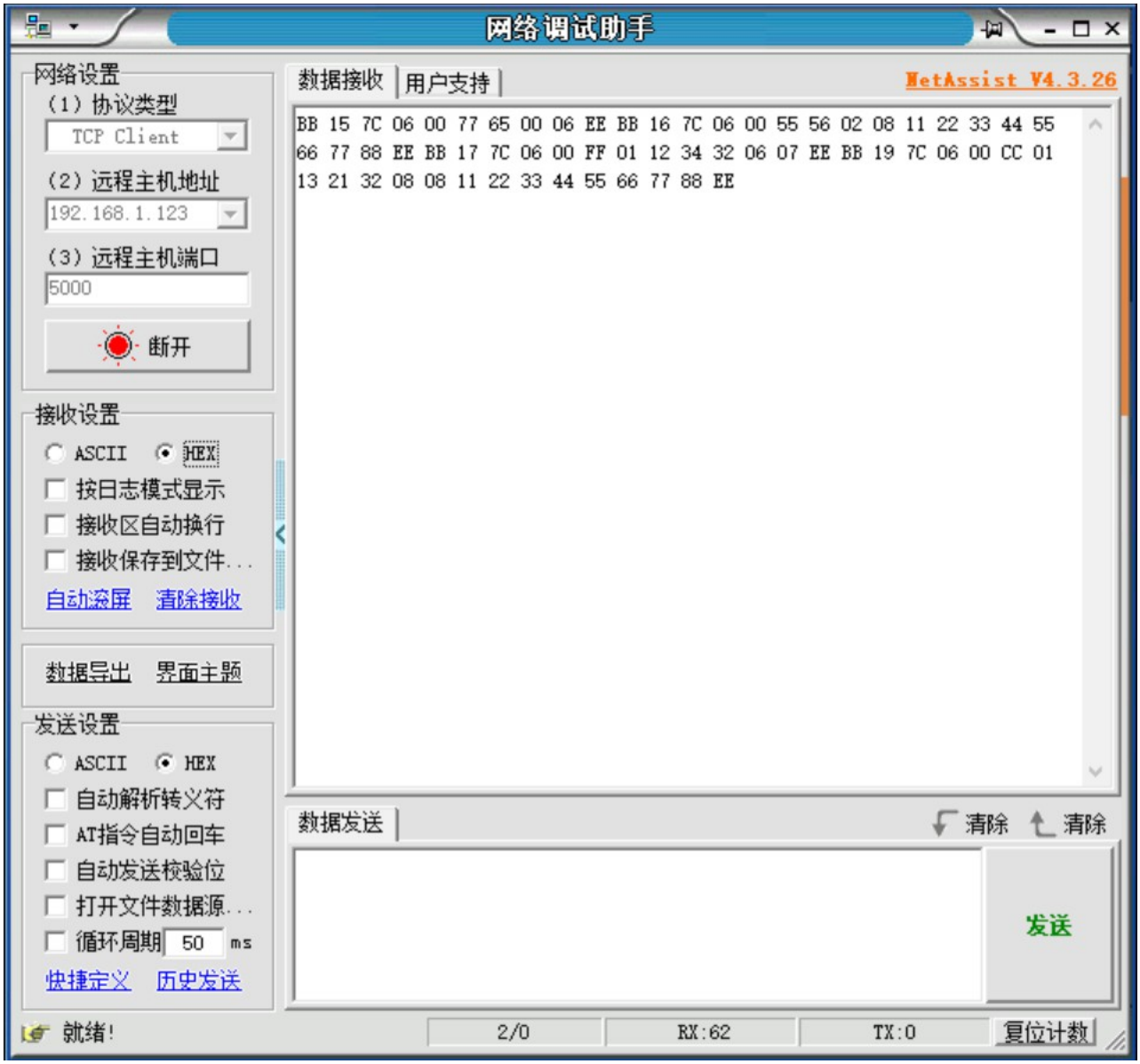


图 16 TTCANopen 工作模块接收指令演示

11. TTCANopen 心跳器模块

用户可以通过配置指令，将设备配置为 TTCANopen 心跳器模块。配置方法见本《说明书》第 8 节。

心跳器模块，是为 TTCANopen 网络提供系统心跳的设备，网络上的设备收到系统心跳会将其本地时间与心跳时间同步，完成总线时间统一。

用户可以将模块配置为 CAN 状态或 CANFD 状态，以使其与用户当前 CAN 网络状态保持一致。

心跳器，完成一次心跳分为两帧发送，第一帧为时间预报帧，第二帧为心跳同步帧，两帧通常间隔 4～5 帧时间。

预报帧：00 0A02 01 19 08 tt tt tt tt re te TT TT

心跳帧：00 0A03 01 19 00

其中：tt tt tt tt 为“预报时间”是紧跟着的“心跳同步帧”的结束时刻，以 0.1ms 为记时单位，0～0x337F97FF 是一天中 0.1ms 的计数个数（最大 863999999），即：所谓 TTCANopen 时间。

re 和 te 是“CAN 接收错误计数”和“CAN 发送错误计数”是 CAN 端口收发错误寄存器中的计数值，是心跳器 CAN 端

口的通讯质量的表征。

TT TT 是“心跳周期”以 ms 为计数单位。默认为 1（E803）秒。

心跳同步帧为固定帧，帧发送结束时刻为时标，网络上的设备，接收此帧后立即将其本地时间值同步到预报时间上，完成与系统时间的同步。

在同一网络中只能有一个心跳器工作，其它心跳器处于备机状态，并将本地时间自动同步到主心跳器上，当检查到主心跳器十次没有发出正常心跳时，备机会自动启动替换主心跳器。

心跳器每发送一次心跳都会在接收窗口显示当前发送的时间。见图 17 所示。

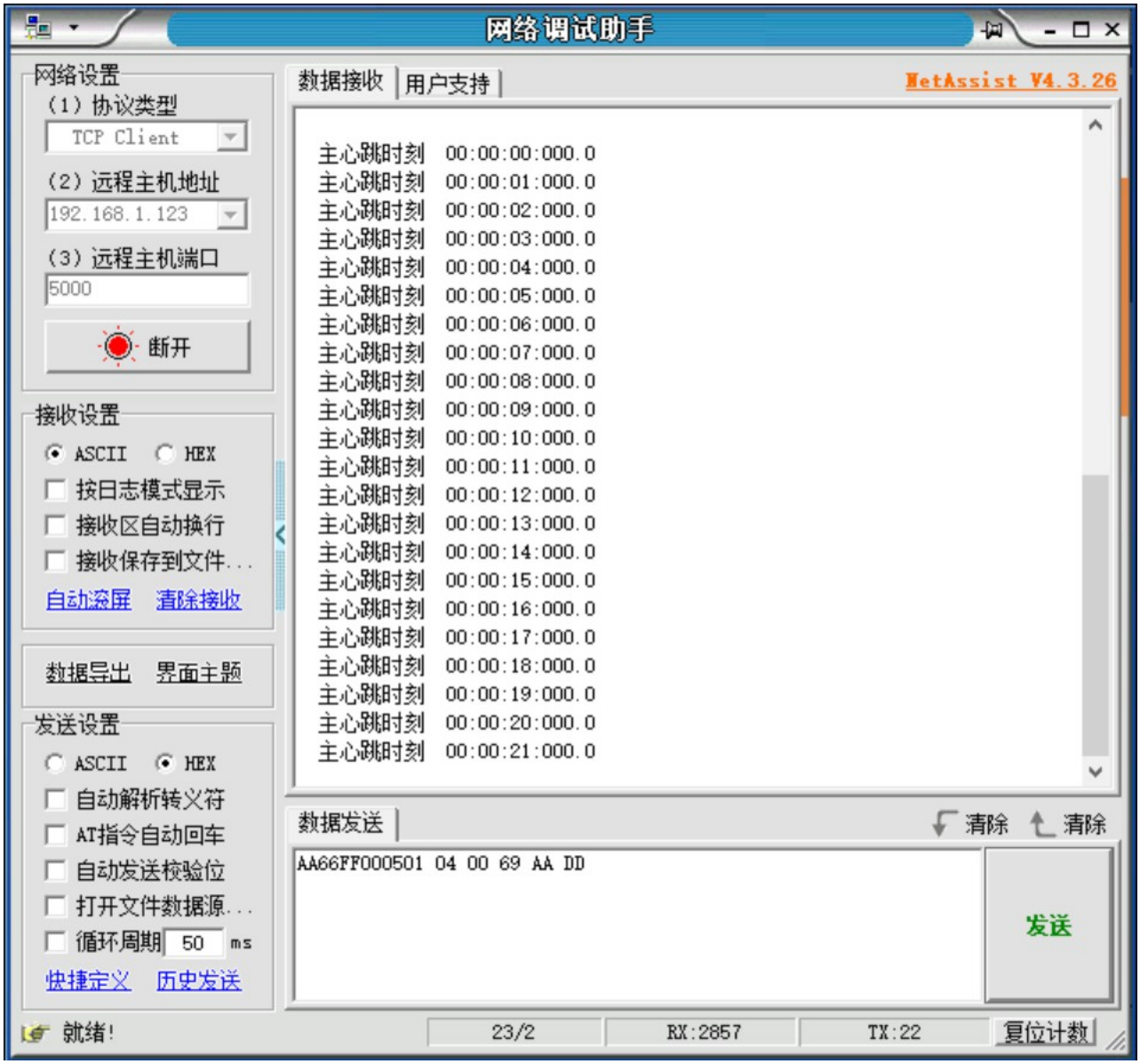


图 17 心跳器显示窗口

用户可以通过发送指令，控制心跳器的启动和停止，也可以设置心跳器的起始时间和心跳周期。

设置发送窗口为字符方式，然后输入？问号并发送，这时在接收窗口会出现帮助信息如图 18 所示。



图 18 心跳器帮助信息

用户可以根据该提示在发送窗口进行所需操作，出错时，会有相关提示。

如果用户需要将心跳周期改为2秒，就可以在发送窗口输入：L02000回车，然后点“发送键”；

如果用户需要将起始时间改为12:12:12:000.0，　可输入：T12:12:12:000.0回车，然后发送。

如果用户需求启动或停止心跳，可输入：Q，　然后发送。

用户也可以发送：N　来结束心跳器工作，恢复设备到调试模块。

12. 模块 TTCANopen 特性（只针对 TTCANopen 状态）

1）模块遵从“TTCANopen 应用层协议 CAN 标识的分配方法”，见附录 1；

2）模块遵从“TTCANopen 应用层协议设备内部多条指令的发送规则”，见附录 2；

3）模块接受系统心跳，并与之同步，图 19 为 TTCANopen 调试模块的时间同步过程。

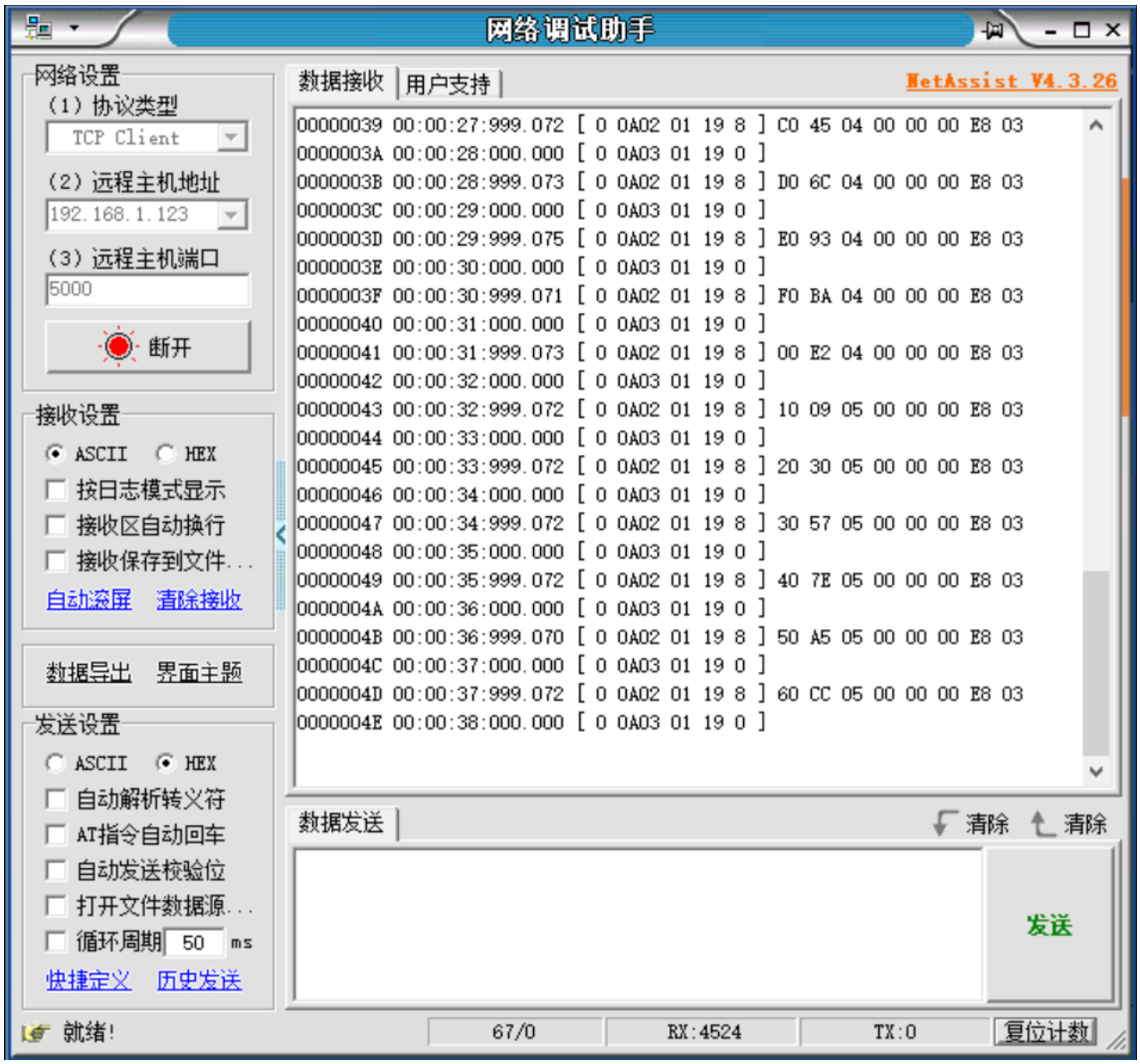
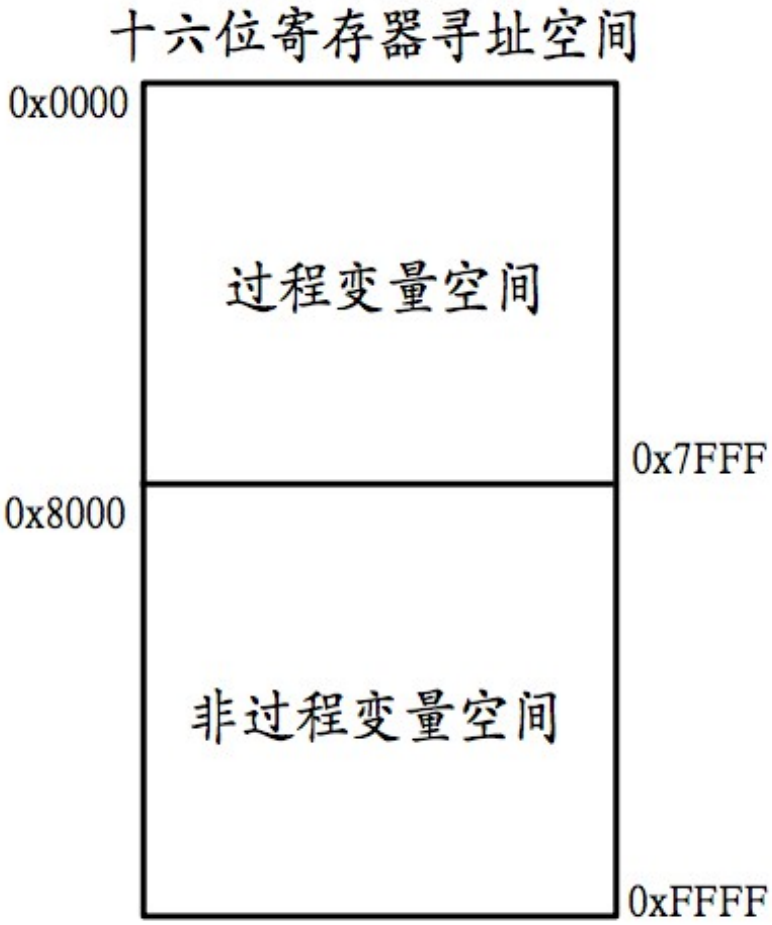


图 19 TTCANopen 调试模块与系统时间同步

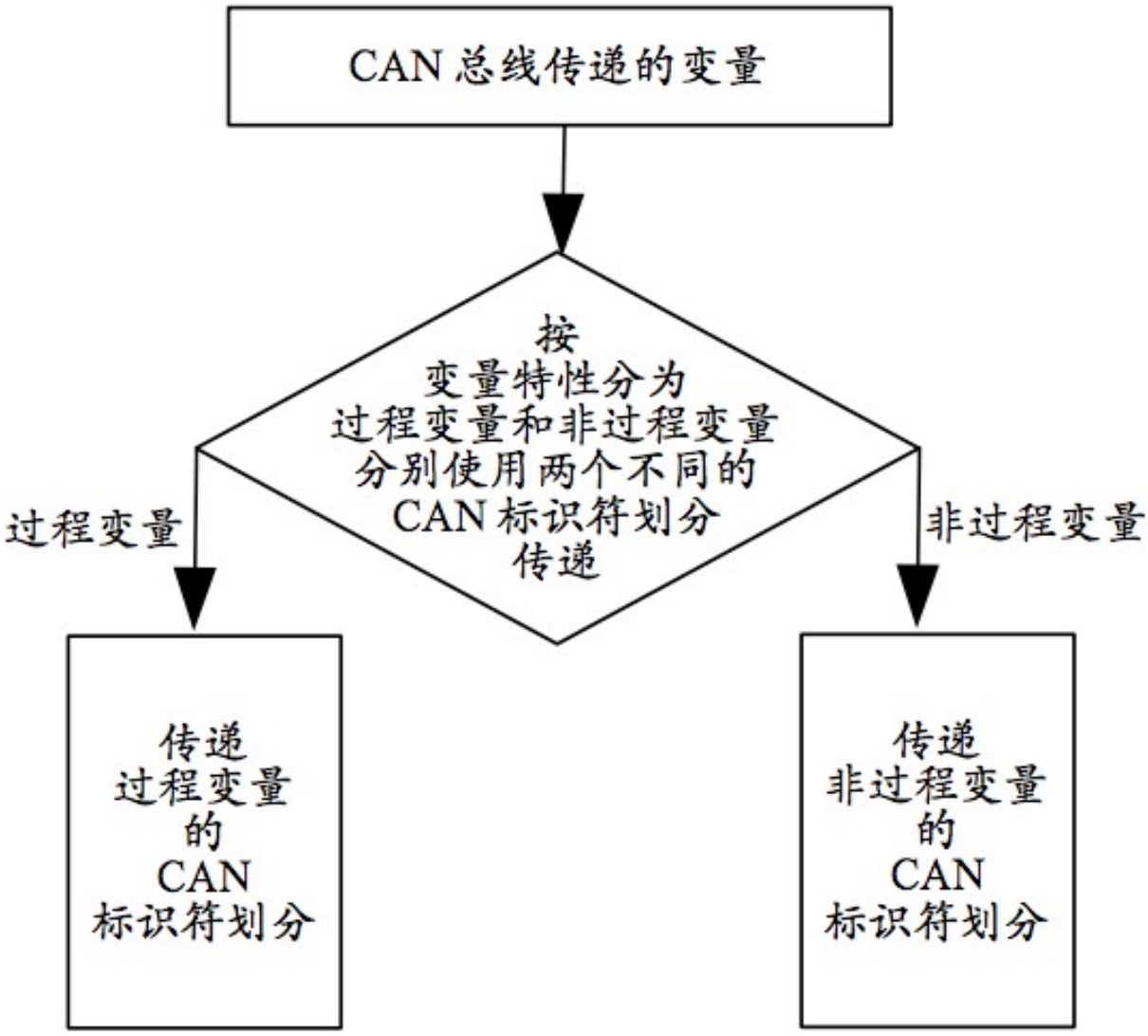
附录 1:　　TTCANopen 应用层协议 CAN 标识的分配方法

TTCANopen 应用层协议以连续寄存器空间为信息载体，并将其十六位寄存器寻址空间平分为两部分，见附图 1 所示。其中：0x0000～0x7FFF 用于存放、传递过程变量；0x8000～0xFFF 用于存放、传递非过程变量。



附图 1 寄存器空间的划分

TTCANopen 应用层协议根据CAN总线传递变量特性的不同，即过程变量或非过程变量，使用两个不同的CAN标识符划分分别传递上述两种变量，见附图2所示，从而为两种变量提供了不同的总线竞争途径，并为应用层协议实施生产者-消费者模型和主机-从机模型提供了明确的对象指向。



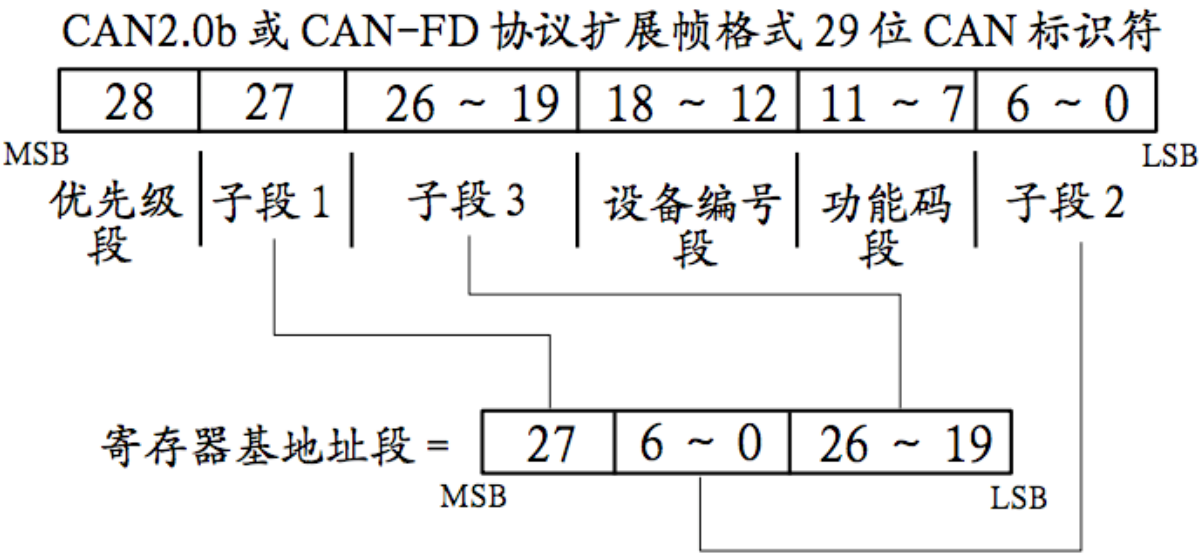
附图2 使用不同的CAN标识符划分传递两种不同的变量

传递过程变量的CAN标识符划分，采用了在CAN标识符中抽取多个位片段，并重新排列拼接组合的方式，构成应用层传递过程变量的寄存器基地址段，从而可以在过程变量寄存器基地址空间构成多个等价竞争层，很好的解决了变量分层管理与总线竞争分配不平衡的矛盾。

传递过程变量的CAN标识符划分，见附图3所示，其将CAN2.0b或CAN_FD协议扩展帧格式中29位CAN标识符划分为六段，即:优先级段、子段1、子段3、设备编号段、功能码段和子段2，其中:子段1、子段2和子段3排列拼接组合构成寄存器基地址段，各段组成说明如下:

优先级段: 由CAN标识符的最高位第28位组成;
子段1: 由CAN标识符的第27位组成;
子段3: 由CAN标识符的第26～19位组成;
设备编号段: 由CAN标识符的第18～12位组成;
功能码段: 由CAN标识符的第11～7位组成;
子段 2: 由CAN标识符的第6～0位组成;

寄存器基地址段:由子段1、子段2和子段3排列拼接组合而成，即:由CAN标识符的第27位、6～0 位和 26～19位组成十六位的寄存器基地址段。



附图 3 传递过程变量的 CAN 标识符划分

当 CAN 标识符的第 27 位(即:寄存器基地址段的最高位)为 0 时，寄存器基地址空间指向过程变量空间(即:0x0000～0x7FFF)，由于十六位寄存器基地址段的低八位(26～19 位)处于 CAN 标识符的较高端 MSB，而其次高七位(6～0 位)处于 CAN 标识符的低端 LSB，由此过程变量寄存器空间变量的竞争能力由其十六位寄存器基地址段的低八位决定，从而在过程变量寄存器地址空间形成了 128 个等价竞争层，每层包含 256 个地址空间，在每层相同位置上的过程变量具有相同的总线竞争能力，在同层中，低地址变量的竞争能力优于高地址变量。

传递非过程变量的 CAN 标识符划分，见附图 4 所示，其将 CAN2.0b 或 CAN_FD 协议扩展帧格式中 29 位 CAN 标识符划分

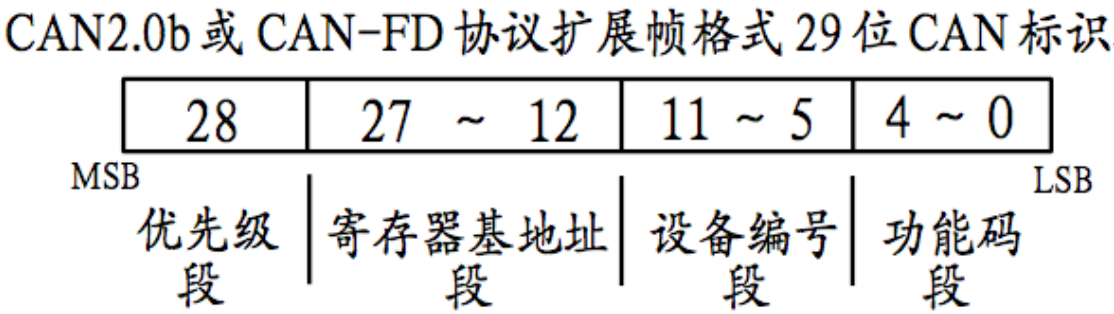
为四段，即:优先级段、寄存器基地址段、设备编号段和功能码段，各段组成说明如下:

优先级段:　　由 CAN 标识符的最高位第 28 位组成;

寄存器基地址段:由 CAN 标识符第 27～12 位组成;

设备编号段:　　由 CAN 标识符的第 11～5 位组成;

功能码段:　　由 CAN 标识符的第 4～0 位组成;



附图 4 传递非过程变量的 CAN 标识符划分

一般非过程变量的实时性要求不高，在应用层通常适用主机-从机模型，主机和从机间采用问答方式通讯，变量分段管理与其总线竞争能力没有突出的矛盾，因此从 CAN 标识符的高端直接截取十六位(27～12 位)构成寄存器基地址段，当 CAN 标识符的第 27 位(即: 寄存器基地址段的最高位)为 1 时，寄存器基地址空间指向非过程变量空间(即:0x8000～0xFFFF)，

非过程变量的总线竞争能力整体低于过程变量，这是应用层所期望的，由于CAN标识符的高七位不能连续出现高电平1，当优先级位(28位)为1时，非过程变量实际有效的地址空间为0x8000～0xFBFF，也就是在0xF000段的尾部集中有1024个地址空间(0xFC00～0xFFFF)不能使用，如果对非过程变量采用和过程变量一样的寄存器拼接组合的方法构成寄存器基地址段，势必将这1024个不可用地址空间分散到各等价层中，这对非过程变量寄存器空间的使用和管理无益，因此应用层对过程变量和非过程变量分别采用不同的CAN标识符划分进行总线传递是非常必要的。

TTCANopen应用层协议对CAN标识符的划分及其物理含义的分配，如:优先级段、寄存器基地址段、设备编号段、功能码段都具有明确的应用层协议指令含义，据此设计的应用层协议具有很好的直读性和开放性，使应用层协议易于解析，便于用户在框架内构建适合其使用环境的应用层子协议。

TTCANopen应用层协议以连续寄存器空间为信息载体，其应用层协议指令是对该连续寄存器空间的读写访问，由此使其能够天然的兼容CAN-FD协议对数据场长度的扩充。

附录 2: TTCANopen 应用层协议
设备内部多条指令的发送规则

通常 CAN 通讯协议只规范了多个设备在总线上同时发送指令时的竞争顺序，而对于单个设备内部产生的多条指令的发送顺序却很少涉及。

在纯主从模式下，设备内部产生多条指令的可能性并不大，相关问题很难暴露，而当系统发展演绎到全触发方式后情况就不同了，一个设备可能具有多个触发变量，它们可以是周期触发、变化触发、越界触发等等，而总线的发送速率是有限的，且同时还存在其他设备对总线的竞争，因此在同一设备中就有可能积累多条指令等待发送。

通常设备会默认以“先生优势”规则处理这些指令，即先产生的指令优先发送，这样处理的最大好处是指令动作先后逻辑规范，这对那些具有先后关联的系列指令尤为重要，但事物总是两方面的，这种“先生优势”规则，会延迟和阻碍后产生的高优先级指令的发送，如：告警指令，尤其是当总线上存在竞争时，情况尤为严重，如:当设备指令发送端口被一条先生成的低优先级指令占据时，其在总线上的竞争处于劣势，从而使该设备中后产生的告警指令无法参与竞争而

被推迟发送，这种延迟有时可能是致命的。 因此我们需要重新规范设备内部多条指令的发送规则，最粗暴简单的方法就是对这多条指令进行优先级排队，将 ID 字小的指令排在前面，每当设备产生一条新指令时都要重新进行一次指令排序(注意:这种重新排序应当包括那条已经处在发送端口的未发指令)，这样设备便具有了最优的发送竞争力，但同时可能会破坏前面所说的某些指令的先后逻辑关联，可能使系统运转出现偏差。

为此 TTCANopen 应用层协议做出如下规范，当一设备同时具有多条指令等待发送时：

1)优先级位为 0 的指令优先于优先级位为 1 的指令发送;

2)优先级位相同的指令按其产生的先后顺序发送。

虽然只有短短的两条规定，但它却完成了其它应用层协议很难规范完成的内容，而这对于 TTCANopen 协议来说却是自然天成的，它的第一条规定确保了低优先级(1)的指令不能阻碍高优先级(0)指令的发送，使设备中诸如告警指令得以顺利抢先通过总线发送；第二条规定保障了具有先后逻辑关系的系列指令的顺利完成，因为我们一般不会将这些相关指令定义在不同的优先级位上。

更多内容请关注：www.ttcanopen.com

免责声明

本着对用户负责的精神，说明书尽可能翔实正确的对产品功能和指标进行描述，但不能保证没有遗漏和错误。另外，随着产品不断升级和完善，说明书内容也具有一定的时效性和适用性，可能会在没有特别通知的情况下，进行修改和补充，请用户及时关注 www.ttcanopen.com 网站中相关信息。感谢您的包容与支持。

补充说明：

1，GD 公司现已经将 GD32E103 系列，更改为 GD32C103 系列。

附录 3:　升级到 v4.0 新增的四项功能

1，**UID 读取指令**，读取到单片机序列号中的一个 32 位数，用户可以通过这个 UID 区分单片机。

指令格式：AA 66 FF 00 01 02 DD

发送该指令后，模块会以字符串形式返回一个 32 位数，

如：26533302

2，配置模块时间指令，在 TTCANopen 应用层协议中，模块时间是由心跳器每秒发送一次主心跳完成总线时间同步的。但对于没有使用 TTCANopen 应用层协议的用户模块，时间是从上电时刻 0 开始计数的。有用户提出需求希望模块时间能够与 PC 机基本同步，为此新增了该指令。

指令格式：AA 66 FF 07 05 01 tt tt tt tt DD

其中， tt tt tt tt 是一个 32 位无符号数，低字节在前高字节在后，是一天中以毫秒为计时单位的时刻值。

用户可以先读取 PC 机当前的时间，并将其转换为毫秒计数，发送该指令，这样就可以将模块的时间配置为与 PC 机基本同步。用户可以通过周期性的发送该指令，去修正同步时间的漂移。

3， **CANFD 数据域不加速发送**，在 v3.7 版本，默认 CANFD 发送数据域是加速的。但在实际应用中，有少数用户在使用数据域不加速的方式发送 CANFD 帧，为了使模块功能更加完备，满足更多用户的应用需求，在 v4.0 版中添加了该功能。

发送方法：在发送指令序列中，将表述 CANFD ID 字的字节序列的最高位置 1.

举例：常规发送指令如：（默认数据域加速）

AA 55 01 23 08 11 22 33 44 55 66 77 88 DD

指令中，01 23 是 ID 字，将 ID 字最高位置 1 后指令为：

AA 55 81 23 08 11 22 33 44 55 66 77 88 DD

再例：

AA CC 12 34 56 78 04 11 22 33 44 DD

指令中，12 34 56 78 是 ID 字，将 ID 字最高位置 1 后指令为：

AA CC 92 34 56 78 04 11 22 33 44 DD

实际发送的 CANFD ID 字，标准帧是 11 位，扩展帧是 29 位的，都没有用到 16 位或 32 位字节 ID 字的最高位，这里最高位置 1，只是用于标识发送不加速，不会影响真实发送的 ID 字。

另外，鉴于不加速帧传输的可靠性更好，模块作为心跳器时，发送心跳帧，使用 CANFD 数据域不加速方式发送。

4，新增6个CAN ID过滤，最初设计USB转CANFD时，由于认为USB端连接的是PC机，将ID过滤留给功能强大的PC机上位机软件去完成比较妥当，后来模块加入了TTL串口、SPI和网口，连接的可能是另一个单片机，尤其是TTL串口，通信速率过低，因此在v4.0版本模块新添加了6个ID过滤。

这6个ID过滤没有设计屏蔽掩码，只是单纯的选则6个ID，具体指令如下：

AA 66 FF 10 18 xx xx xx xx yy yy yy yy zz zz zz zz mm mm mm mm nn nn nn nn kk kk kk kk DD

当然您也可以配置少于6个ID过滤，如：只配置2个ID过滤，

AA 66 FF 10 08 xx xx xx xx yy yy yy yy DD

其中，xx xx xx xx 和 yy yy yy yy 等，都是低字节在前高字节在后的无符号32位数，代表需要选取的指定ID值，这样模块就会只传递这些指定ID的帧，其它被忽略。

如：您只想看ID字等于0123的帧，就需要发送配置指令：

AA 66 FF 10 04 23 01 00 00 DD

注意：无论是标准帧还是扩展帧都使用32位过滤字，在上面的指令发出后，您可以看到ID字是0123的标准帧，也可以看到ID字是00000123的扩展帧。

另外，需要注意的是过滤ID字必须从地址FF 10开始配置起，而且第一个过滤ID字不能是0（即xx xx xx xx不能是0）。

如果第一个过滤 ID 字是 0，模块将忽略 ID 过滤功能，将显示全部所收到的帧。